CCP Application Integration Thread

Atlas Design Panel 1

February 24,
1998

84K00303-003
Basic

CCP Application Integration Thread
Table of Contents

Assessment Team

Name	CI Represented	E-Mail Address	Phone
John Copella	Thread Lead	copelj%kscgws00@ksc.nasa.gov	861-9087, 800-709-4581
Rich Ikerd	Real Time Control (RTC) Applications	X.500	861-7541
Robert Pierce	RTC Applications	X.500	861-7505
Ken Hale	RTC Applications	X.500	861-2296
Jim Trawick	Application Services (Tailored)	trawijc%kscgws00@ksc.nasa.gov	861-9042, 800-592-3036
Julia Samson	Application Services (Basic)	X.500	861-9001
Rodney Davis	Constraint Management	X.500	861-9082
Walter Clavette	Command Support	X.500	861-9043
Gary Hrezo	System Viewers	hrezogw%kscgws00@ksc.nasa.gov	861-9002
Steve Moore	OPSCM	moorecs@kscgws00.ksc.nasa.gov	861-9203
Bob McMahon	System Services	X.500	861-9045
Al Menendez	System Build	X.500	861-2204
Scott Estes	Simulation	X.500	861-2403
Oscar Brooks	Operating Systems	X.500	861-7232
Charla King	Test Build	X.500	861-7587

1. INTRODUCTION

1.1 OVERVIEW

This thread continues with establishing the CLCS capability to support application software on the Command and Control Processor (CCP). This includes providing the services required to support: End Item Manager (EIM), Prerequisite Control Logic (PCL), and Test Application Script (TAS) application types. It also includes the interfaces required to enable communication between: EIM and TAS, EIM and PCL, and EIM and Test Application Displays, which reside on the Command and Control Workstation (CCWS).

The primary purpose of this thread is to get the End Item Manager to full working status in the RTPS system. A secondary objective is to demonstrate combined functionality of TAS and EIM applications.

1.2 CONCEPT

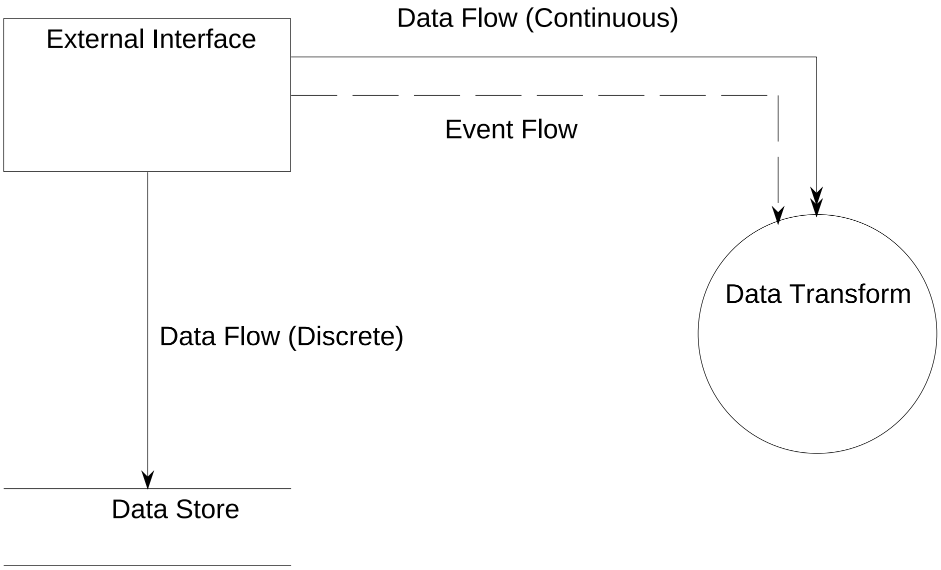
The focus of this thread is to continue development of required Application Software interfaces per the Application Software API requirements matrix (Attachment 1), enable integration of Application Software with the provided interfaces, and to ensure that the combined system results in the desired functionality. Simulation resources (HMF math models) will be used to develop and test the capabilities delivered in this thread.

Core interfaces to the RTPS are provided by Basic Application Services, which is delivered as a set of C/C++ header files and libraries. The intent of Basic Application Services is to present a consistent set of interfaces to RTPS resources that are required by all types of applications. Basic Application Services is designed to present a consistent, vendor-independent framework for application development while insulating applications from changes in the underlying RTPS architecture, operating systems, and hardware.

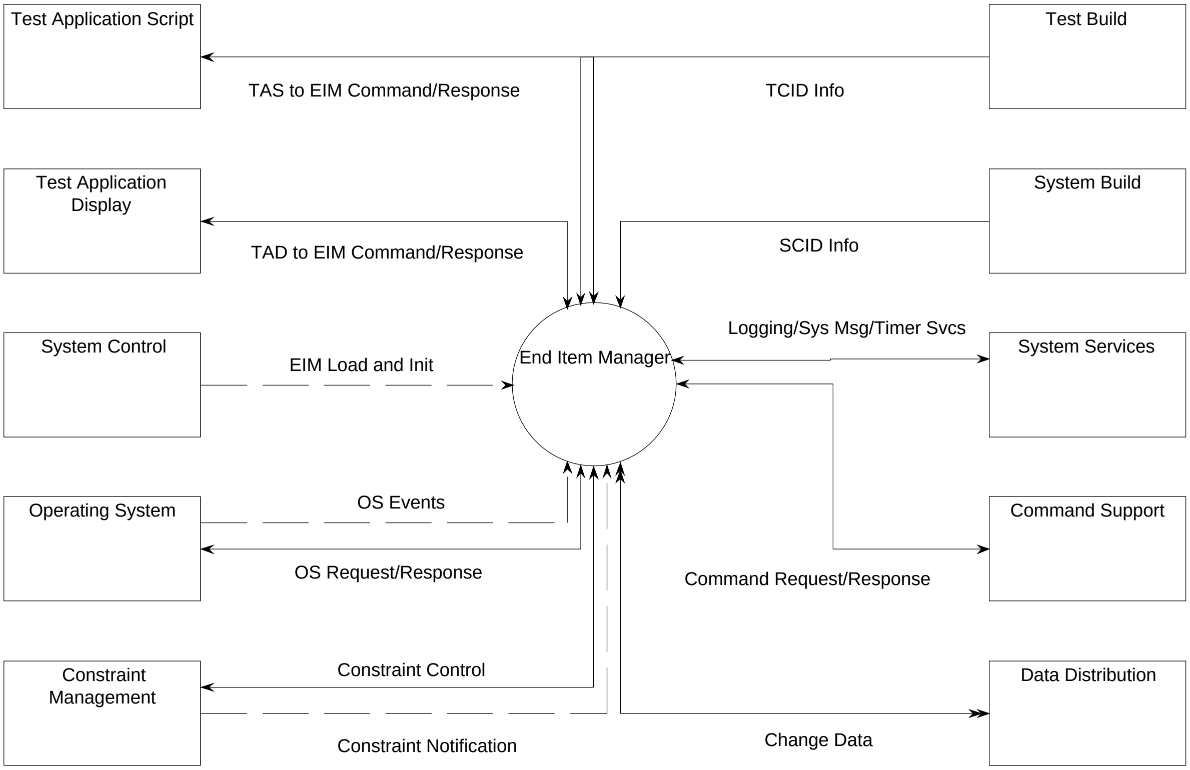
Each of the different application types interfaces directly with a specific CSC provided by Tailored Application Services. EIMs employ EIM Services, PCL applications use PCL Services, and TAS requires TAS Services. Each Tailored Application Service CSC exposes from Basic Application Services selected interfaces (APIs) appropriate to each application type. Each Tailored Service is delivered in the form of a set of C/C++ headers, libraries, and, for those application types built using a commercial tool, vendor-specific components that enable applications built with the tool to interface with the RTPS.

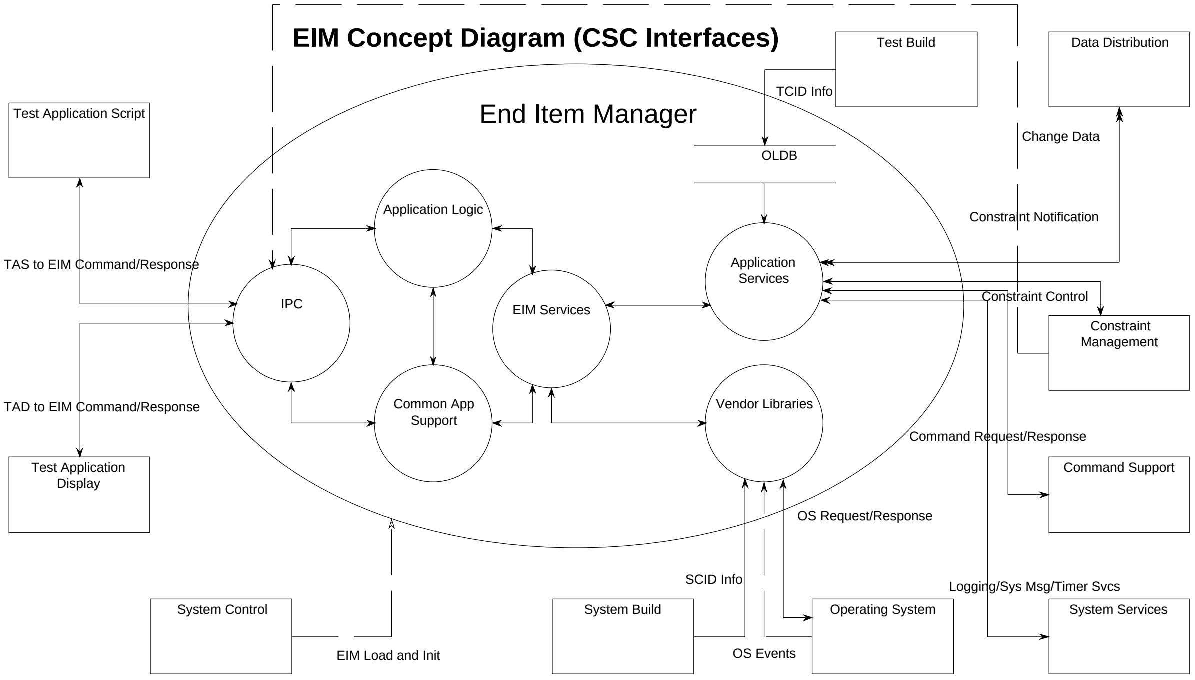
The Interprocess Communication (IPC) CSC of System Services provides the technology that allows EIM, TAS, and Test Application Displays to communicate with each other over user-defined interfaces.

Diagram Legend



End Item Manager Level 0 Context Diagram





EIM Concept Diagram (CSC Interfaces)

End Item Manager

Test Application Script

TAS to EIM Command/Response

TAD to EIM Command/Response

Test Application Display

IPC

Application Logic

Common App Support

EIM Services

Application Services

Vendor Libraries

OLDB

Test Build

TCID Info

Data Distribution

Change Data

Constraint Notification

Constraint Control

Constraint Management

Command Request/Response

Command Support

OS Request/Response

Logging/Sys Msg/Timer Svcs

System Services

SCID Info

System Control

EIM Load and Init

System Build

OS Events

Operating System

1.3 CCP APPLICATION INTEGRATION THREAD SPECIFICATION

1.3.1 Statement of Work

1.3.1.1 Thor Carry-over

This section is the statement of work from the Thor EIM and Application Interface Thread. Since many of the Thor SOW items were open-ended, they remain in the document. Work items that were italicized in the Thor statement of work and will be worked during this thread are preceded by "**(Atlas)**".

- The design/implementation will use an object-oriented paradigm to mesh with the Application Services and Applications Software design.
- Utilize inter-process object-based communication capability to directly invoke a remote object's methods.
- Provide an interface from the application to Command Management to prevent unauthorized command attempts from leaving the CCP.
- Provide the capability to operate an End-Item using rate based control.
- Provide an interface to allow End Item Manager Tool (Control Shell) to:
  - Read and write all Thor-delivered FD's and End Item Manager Components and their attributes.
  - Receive control commands from users on an CC W/S.
  - **(Atlas) Receive control commands from Test Application Scripts (this statement of work item will be revised and expanded in the TAS DP2 after the Thor TAS Pathfinder demonstration in late Feb/early March 1998).**
  - Delegate continuous monitoring to Constraint Management and respond to Constraint Management notification events.
  - Issue a command to another End-Item Manager and receive a response.
- **(Atlas) Provide priority Reactive Sequence Control processing for End-Item Managers that requires minimum reaction time to event notification from Constraint Management**
- Provide the interfaces to ensure that Prerequisite Control Logic (PCL) processing occurs for all issued FD commands that have associated PCL applications, regardless of the origination of the command (automated application FD command issuance or manual FD command issuance)that:
  - Allows Command Management to route issued FD commands to Prerequisite Control Logic
  - Execute Prerequisite Control Logic unless specifically overridden
  - Inhibit execution of the command if the Prerequisite Control Logic condition(s) are not met.
  - Provide results of Prerequisite Control Logic to Command Management .
  - Issue System Message if Prerequisite Control Logic condition(s) are not met.
  - Provide an override to Prerequisite Control Logic for CC W/S command processor.
  - Provide a method to list which command FD's have Prerequisite Control Logic.
  - Provide a method for overriding an FD input to a Prerequisite Control Logic application.
- Ensure system level compatibility between End Item Managers, FD's, Test Application Displays, and Prerequisite Control Logic. Define checking and binding at compile, build, startup, and runtime.
- **(Atlas) Provide performance data for system modeling**.
- Provide the capability for End Item Manager to be utilized in ~~both~~ Operational ~~and Desktop Debug~~ configurations. **(Desktop Debug Environment (DDE) related impacts assessed in DDE thread)**

- Provide logging of error, performance and state change information
- Baseline system messages using the System Message Catalog to include message and help text.

1.3.1.2 New Functionality for Atlas

1.3.1.2.1 ControlShell™ Version 6.0 Integration

1. Procure, install, and integrate ControlShell ™ v6.0 into the CLCS development network. A beta release will be delivered in late January with the production software to follow approx. 8 weeks later (Operating Systems, System Build, EIM Services). ControlShell™ requires the following freely available products to be installed on the host (development) platform:
   - Java Development Kit (JDK) v1.1.3 (www.java.sun.com)
   - JGL v2.0 (Java GUI widget library, available from www.objectspace.com)
   - Perl v5.002 or higher (prep.ai.mit.edu)
2. Recommend and implement, if possible, a solution for the excessive compile/link times reported for EIM applications. (System Build, Operating Systems, CLCS Network Support). Refer to Attachment 3 for details.
3. Develop procurement and installation plan for operational (OCR-resident) version of ControlShell ™. The operational version provides control (stop/step/goto) and visibility into a running application but does not permit the application to be modified. The operational version is known informally as the "read only" product (RTC Applications, SEI, EIM Services [Dwayne Burns]).
4. Acquire vendor training for the v6.0 product, and begin development of KSC tailored training materials (RTC Applications, EIM Services, KSC Technical Training).
5. Update EIM Services software to compatibility with the ControlShell™ v6.0 production release (EIM Services).
6. Update EIM Services software to compatibility with Thor Application Services. To greatest practical extent, maintain compatibility with Atlas Application Services as they are developed (EIM Services).
7. Provide and populate configuration-controlled ControlShell™ repository with EIM Services components (EIM Services).
8. Provide EIM Services user documentation and example code (EIM Services).
9. Provide and begin population of configuration-controlled ControlShell™ repository for reusable EIM application components (Common Application Support).
10. Obtain and compare labor metrics for actual effort incurred to develop EIM software with estimates from NLPS "60-day" effort (RTC Applications).
11. Provide technical evaluation of ControlShell™ v6.0 production software (RTC Applications).

1.3.1.2.2 General

1. Implement Application Services identified for development during Atlas per the Application Software API requirements matrix (Application Services [Basic and Tailored]), incorporating "lessons learned" during Thor EIM Services integration.
2. Provide example-based documentation for EIM developers that is keyed to API requirements matrix (Application Services [Basic and Tailored]). The intent of this item is not to create a new document, but to refine and improve the usability of existing documentation products.
3. Provide application level support for enumerated data types, function designators (FDs) and pseudo FDs (DBSAFE, Test Build, Application Services, EIM Services).

4. Provide application level support for string data types and pseudo FDs (Test Build, Application Services, EIM Services).
5. Develop standardized interfaces between EIM and Test Application Displays for the following functions (Common Application Support):
   - Sequencer option selection
   - Sequencer start, stop, “single step”, “go to”, resume, secure, and terminate
   - Sequencer prompt/response
   - Sequencer progress reporting
   - Sequencer test results reporting
6. (Info only – this SOW item is covered by Test Build and Load Phase 2 Thread) Continue incorporation of EIM application and services software into the Test Build process (Application Software, EIM Services, Test Build).
7. (Info only – this SOW item moved to Redundancy Management Phase 1 thread) Provide input/feedback regarding operations concept and SLS level requirements for application fail-safe and fail-operational capabilities (RTC Applications).
8. Identify requirements and provide initial design for detection of, and recovery from, failures in communications cycle between EIM and Test Application Display applications (RTC Applications, System Services).
9. Develop EIM application component documentation and example code (Common Application Support).
10. Develop configuration management and Test Build integration plans for EIM applications component repository (Common Application Support, Test Build).
11. Develop interfaces required to enable high-priority processing of constraint violations associated with Reactive Control Logic (RCL) (Constraint Management, EIM Services, System Services, ref. SE&I action item #TBD [Change RCL latency requirement baseline to 500 msec]).
12. Refine and document technique for EIM application requirements capture and EIM design (RTC Applications). (SE&I has the action to determine if this SOW item should be moved to the RTC Applications SDP)
13. Develop and document a technique for translating an object-oriented analysis/design model (developed with Paradigm Plus or a similar tool) to a ControlShell™ implementation (RTC Applications). (SE&I has the action to determine if this SOW item should be moved to the RTC Applications SDP)
14. Investigate feasibility of automatic generation of End Item Components (EIC) or EIC interfaces directly from requirements data products (RTC Applications).
15. Develop application software necessary to verify/validate delivered external interfaces (EIM Services).
16. Develop application software necessary to acquire EIM-related performance benchmarks (EIM Services).

1.3.2 Requirements

(SLS 2.2.3.3.3) The CLCS shall provide the capability to restrict issuance of commands to only authorized users and applications [Complete].

(SLS 2.2.5.1.1) The RTPS shall provide fault tolerant End-Item monitoring and control.

(SLS 2.2.5.1.2) The RTPS shall provide End-Item monitoring and control definition that is directly understandable in form and content by End-Item engineering personnel [Complete] [HMF].

(SLS 2.2.5.1.3) The RTPS shall provide a layered technique for defining End-Item monitoring and control that supports reuse and reduction of time to implement [Complete] [HMF].

(SLS 2.2.5.1.4) The RTPS shall provide the capability to monitor all End-Item FD's current state against expected state during all operational End-Item phases [Complete] [HMF].

(SLS 2.2.5.1.5) The CLCS shall provide effective support to test director and test management personnel to clearly understand End-Item and test summary status [Low].

(SLS 2.2.5.1.6) The CLCS shall provide effective support to test director and test management personnel to efficiently and reliably control and track testing progress [Low].

(SLS 2.2.5.1.7) CLCS shall provide the user the capability to create Pseudo Measurement FDs and generate their values using Test Applications [Complete] [HMF].

(SLS 2.2.5.1.8) CLCS shall provide the capability for Pseudo Measurement FD values to be persistent between different tests [Complete] [HMF] (info only, requirement allocated to Checkpoint/Restart thread).

(SLS 2.2.5.1.9) CLCS shall provide the user the capability to choose which Pseudo Measurement FDs are to be persistent [Complete] [HMF] (info only, requirement allocated to Checkpoint/Restart thread).

(SLS 2.2.5.1.10) CLCS shall provide a set of System Status Function Designators which depict the status of subsystems within an RTPS [Medium] [HMF].

(SLS 2.2.5.1.11) CLCS shall support a set of System Function Designators which depict information about the RTPS and the test in progress (e.g., GMTCDT, Date, CDT, GW, TCID Name) [High] [HMF].

(SLS 2.2.5.4) Constraint Management

(SLS 2.2.5.4.2) CLCS shall provide the capability for multiple (TBD number) users and system or user applications to request notification of constraint events for each Measurement FD [Complete] [HMF].

(SLS 2.2.5.4.3) CLCS shall provide the capability for each user, and system or user application requesting constraint notification to specify the limits/condition under which they will be notified [Complete] [HMF].

(SLS 2.2.5.5) Prerequisite Control Logic

"To protect against inadvertently issuing a command which could injure personnel or damage equipment, CLCS provides the capability for the user to predefine logic that verifies the appropriate conditions are met prior to executing the command".

(SLS 2.2.5.5.1) RTPS shall provide the capability to verify that certain conditions are met before issuing any FD command to an End-Item [Complete] [HMF].

(SLS 2.2.5.5.2) The RTPS Prerequisite Control Logic function shall be fault tolerant.

(SLS 2.2.5.5.3) RTPS shall provide the capability to manually override Prerequisite Control Logic once command issuance has been blocked [Complete] [HMF].

(SLS 2.2.5.5.4) RTPS shall provide the capability for Reactive Sequence Test applications to bypass Prerequisite Control Logic [Complete] [HMF].

(SLS 2.2.5.6) End-Item Management

"The End-Item Management function provides the capability for users to create End-Item Manager applications to perform closed loop control of a specific End-Item system or component. A hierarchy of End-Item Managers might exist for large and complex End-Item Systems".

(SLS 2.2.5.6.1) RTPS shall provide the capability to delegate and define closed loop control of one or more End-Items to an End-Item Manager [Complete] [HMF].

(SLS 2.2.5.6.2) The RTPS End-Item Manager function shall provide the capability to automatically safe the End-Item and continue if commanded when a RTPS system fault occurs.

(SLS 2.2.5.6.3) RTPS shall provide the capability to delegate continuous End-Item monitoring to Constraint Management and respond to Constraint Management notification events [Complete] [HMF].

(SLS 2.2.5.6.4) RTPS shall provide the capability to operate an End-Item using rate based control [Complete] [HMF].

(SLS 2.2.5.6.5) RTPS shall provide the capability for an End-Item Manager Test Application to issue a command to another End-Item Manager Test Application [Complete] [HMF].

(SLS 2.2.5.6.6) RTPS shall provide the capability to alter the state of frame rate/time domain control based on an event [Complete] [HMF].

(SLS 2.2.5.6.7) RTPS shall provide priority Reactive Sequence control processing for End-Item Managers requiring minimum reaction time to Constraint Management notification events [Complete] [HMF].

(SLS 2.2.5.6.8) RTPS shall provide an End-Item control layer/encapsulation allowing maintenance in a single place and reuse throughout user applications [Complete] [HMF].

Miscellaneous System Services:

(SLS 2.2.10.1.16) The CLCS shall provide the capability for System and User Applications to record data to the SDC [Application compliance with requirement will be complete in Atlas, service requirement is allocated to Record/Retrieval and Checkpoint/Restart threads] [HMF].

(SLS 2.2.10.1.17) The RTPS shall provide the capability to System and User Applications to record limited messages on the local computer's hard file storage device [Application compliance with requirement will be complete in Atlas, service requirement is allocated to Record/Retrieval and Checkpoint/Restart threads] [HMF].

(API Matrix 4.6.1) The system shall provide a method to identify system errors caused by API calls [Complete] [HMF].

(API Matrix 4.6.2) The system shall provide a method to translate API error codes into system text messages which may be displayed for the user [Complete] [HMF].

(API Matrix 4.6.3) The system shall provide a method to specify application error routines which are called on the occurrence of user specified error conditions [Complete] [HMF].

(API Matrix 4.7.1.1) The system shall provide a method to specify timer events which can occur at user defined intervals with a granularity of 10 milliseconds [Complete] [HMF].

(API Matrix 4.7.1.2) The system shall provide a method to specify multiple timers within a single application [Complete] [HMF].

(API Matrix 4.9.6) The system shall provide a method for opening, closing, updating, reading, and writing disk files in a user definable format [Complete] [HMF].

1.4 CCP APPLICATION INTEGRATION THREAD HARDWARE DIAGRAM

None.

1.5 CCP APPLICATION INTEGRATION THREAD DELIVERABLES

Deliverable	R&D Document	Code	API Manual	Users Guide
Application Services [Basic]		X	X	X
Application Services [Tailored]		X	X	X
Common Application Support	X	X		X
RTC Applications	X			

The following are additional “non CSCI” deliverables from the CCP Application Integration Thread:

1. Performance data for system modeling.
2. Updates to System Message Catalog as required.
3. ControlShell “read only” procurement and installation plan.
4. ControlShell technical Evaluation report.
5. Labor metric report (for EIM development).
6. RTC Application Software practices that address requirements capture and EIM requirements analysis (depending on outcome of SE&I action – refer to SOW section).
7. Feasibility report on automatic generation of End Item Components from requirements data products.

1.6 CCP APPLICATION INTEGRATION THREAD ASSESSMENT SUMMARY

1.6.1 Labor Assessments

The total Labor Costs required to provide this capability are summarized in the following table:

No.	CSCI/HWCI Name	Atlas LM	Changes covered in
1	Application Services CSCI (Basic) • FD Services • Subsystem Services • Onboard Services • Constraint Management Services	8.0	CCP Application Integration Thread Commanding & CPRO Phase 3 Thread Desktop Debug Env. Phase 2 Thread
2	Application Services CSCI (Tailored) • End Item Manager Services • Prerequisite Control Logic Services • Test Application Script Services	 48.0 12.0 24.0	CCP Application Integration Thread
3	Command Support CSCI • Command Processor • Command Management • Authentication Services	0.0	Commanding & CPRO Phase 3 Thread
4	Data Distribution Processing CSCI • Data Distribution • Data Fusion • Data Health • Constraint Management	0.0	System Data and Routing Thread Constraint Manager Completion Thread
5	System Control CSCI • Ops Configuration Manager • RTPS System SW Load and Initialization • Test Load	0.25	CCP Application Integration Thread System Control Phase 1 Thread
6	System Services CSCI • Logging Services • Interprocess Communication Services • System Message Services • Timer Services • Initialization & Termination Svcs	0.0	System Data and Routing Thread
7	System Viewers CSCI • FD Viewer • PCL Viewer • Control and Navigation System (CNS)	5.0	CCP Application Integration Thread CCWS Phase 1 Thread
8	Common Applications Support • EIC Repository • Sequencer/Display Common Interfaces • Test Build integration plans	8.0	CCP Application Integration Thread HMF IPT Vehicle Power Up/Down IPT Orbiter/SRB Hydraulics IPT Cryo/MPS/FIREX IPT
9	Operating Systems	TBD	CCP Application Integration Thread
10	System Build	0.0	System Build, Platform Build and Load Phase 2 Thread
11	TCID Build & Control	0.0	Test Build and Load Phase 2 Thread
	TOTAL	**105.25**	

1.6.2 Hardware Costs

None.

1.6.3 CCP Application Integration Thread Procurement

An additional 20 licenses for the Dynamic Object Oriented Requirements Systems (DOORS) will be procured during Atlas. Tool selection criteria and justification are specified in Attachment 2. DOORS will be used for application software requirements capture, overview specifications, test plans, and requirements traceability. The project has already procured 10 licenses.

A summary of the procurement schedule is shown below:

Procurement Activity	Completion Date
Submit purchase request to procurement	2/15/98
Receive licenses	3/15/98

During Atlas, an additional 25 development seats for ControlShell ™ v6.0 will need to be procured, for an approximate cost of $200,812. The project currently owns 33 licenses for the ControlShell™ product. The procurement schedule for these licenses has not yet been defined.

Also during Atlas, the procurement and installation plan for the operational ("read only") version of ControlShell™ will be established. The procurement schedule for these licenses has not yet been defined.

1.7 CCP APPLICATION INTEGRATION THREAD SCHEDULE & DEPENDENCIES

1.7.1 Schedule

Concept Phase	**1/14/98**	**3/13/98**
Concept Definition	1/14/98	1/30/98
Concept Panel Preparation	1/21/98	1/30/98
Concept Panel Flip Through	N/A	N/A
Preliminary Concept Panel		2/3/98
Concept Panel Rework		2/4/98
Concept Panel		2/5/98
Delta Concept Panel		3/12/98
Work off Concept Panel Action Items	2/6/98	3/13/98
Requirements Phase	**1/21/98**	**3/27/98**
Refine Requirements	1/21/98	3/2/98
Requirements Panel Preparation	3/2/98	3/12/98
Requirements Panel Flip Through		3/12/98
Preliminary Requirements Panel		3/17/98
Requirements Panel Rework		3/18/98
Requirements Panel		3/19/98
Work off Requirements Panel Action Items	3/20/98	3/27/98
Preliminary Design Phase	**2/6/98**	**4/24/98**
Preliminary Design	2/6/98	4/2/98
Design Panel Preparation	4/2/98	4/9/98
Design Panel Flip Through		4/9/98
Preliminary Design Panel		4/14/98
Design Panel Rework		4/15/98
Design Panel		4/16/98
Work off Design Panel Action Items	4/17/98	4/24/98
Detail Design Phase	**3/23/98**	**4/15/98**
Code and Test Phase	**4/15/98**	**9/25/98**

Code and Unit Test	4/15/98	6/15/98
Integrate and Test	6/15/98	8/15/98
Support System Test	8/15/98	9/25/98

Thor Delivery		**3/27/98**
Atlas Delivery		**9/25/98**

1.7.2 Dependencies

No.	Dependency Area	Dependency	Need Date	Possible Mitigation
1	EIM Services RTC Applications	Control Shell 6.0 Licenses	3/27/98	Use 6.0 Beta
2	EIM Services RTC Applications	Data Retrieval Capability	6/1/98	Use local logging
3	Application Services [Basic] EIM Services RTC Applications	Simulation Connectivity	3/27/98	None

1.8 CCP APPLICATION INTEGRATION THREAD SIMULATION REQUIREMENTS

Connectivity to SGOS math models (either via VSI or Simulation Gateway) will be required for test. Math model support for all Atlas-delivered FD types will be required for test.

1.9 CCP APPLICATION INTEGRATION THREAD INTEGRATION AND SYSTEM TEST

A test bed configuration will be created using the HMF TCID. An additional test TCID will be required in order to exercise LDB interfaces, since the HMF TCID is GSE-only. Applications will be developed in Control Shell™ and Sherrill Lubinski (SL) by EIM Services that exercise all of the interfaces required to implement the CCP Application Integration Thread.

The objective of the tests is to demonstrate System Level Compatibility between End Item Manager and its external interfaces. In addition, a sample Test Application Script will be developed, and the ability of the script to send a command and receive a response from an EIM will be demonstrated.

1. EIM, Test Application Display, TAS, and PCL Applications will be developed that exercise all of the options developed for this thread in Thor.
2. The EIM, Test Application Display, TAS, and PCL Applications will be compiled, built into the TCID, and brought on-line into the SDE and IDE.
   - Error free compile/link will be verified.
   - Proper loading of the EIM, Test Application Display, TAS, and PCL Applications to the CCP & CCWS will be verified.
3. The EIM/Test Application Display Application Sets will be executed and each function will be exercised. This will verify:
   - Proper operation of CORBA I/F between Test Application Displays and EIM (Control Shell™)
   - EIM Application Interfaces with Test Application Displays, PCL, Command Management, Command Authentication, Constraint Management, and System Message processing.
4. The EIM/Test Application Script Application Sets will be executed and each function will be exercised. This will verify:
   - Proper operation of I/F between Test Application Script and EIM (Control Shell™)
5. Retrieval programs will be run to demonstrate proper recording of Significant Events

- EIM Initiation and Termination.
- Command Issuance, Authentication Failure, PCL Execution, Constraint Checking
- System Messages

6. Commands to the EIM will be attempted from CCWS that are not authorized to control the EIM and verified that the attempt fails.
7. Tests 2 (loading only), and 3 will be run with the EIM, Test Application Displays, & PCL loaded and executing in the Desktop Debug Environment. This will verify the proper execution of these products in the Desktop Debug Environment.

1.10 CCP APPLICATION INTEGRATION THREAD TRAINING REQUIREMENTS

Assessments for DOORS and ControlShell™ training are contained in this thread.

1.10.1 Training Needed

Application Services (Tailored) requires ControlShell training for 6 engineers.

Application Services (Tailored) requires DOORS training for 2 engineers.

Application Services (Basic) requires DOORS training for 2 engineers.

RTC Applications requires ControlShell training for 10 Software Engineers.

RTC Applications requires DOORS training for 10 Software Engineers, and 10 Shuttle Systems Engineers.

Summary:

ControlShell v6.0:	16 people (6 ASV, 10 RTC)
DOORS:	22 people (2 ASV, 20 RTC)

1.10.2 Training to be provided

The required training will be provided during this thread.

1.11 CCP APPLICATION INTEGRATION THREAD FACILITIES REQUIREMENTS

Connectivity between SGOS math models and the SDE/IDE will be required for development testing. Either the existing VSI or Simulation Gateway interface is acceptable.

1.12 TRAVEL REQUIREMENTS

Four people for 2 days/1 night at Real-Time Innovations, Inc. (RTI) facilities in Sunnyvale, CA. Suggested time frame is mid-July. Attendees should include one individual from the following groups:

1. RTC Applications Software
2. SE&I Technical Staff
3. SE&I Management
4. Project Management (to discuss business relationship, contracts, and pricing)

Objectives of the visit are to:

1. Present project status and our EIM design to their technical staff
2. Review RTI product plans for the upcoming year
3. Conduct technical interchange
4. Conduct dialog on how RTI can further assist NASA with the project

1.13 CCP APPLICATION INTEGRATION THREAD ACTION ITEMS/RESOLUTION

1. Develop plan for ControlShell™ "read only" product.
2. Provide procurement schedule for additional ControlShell™ licenses.
3. Need to identify detailed OS support requirements for integration of ControlShell™ v6.0. Oscar Brooks & Operating Systems group should assess the package. (in work/awaiting feedback from Oscar)
4. RTC Application deliverables for requirements capture and OOD to ControlShell™ mapping may not belong in this thread (should be in RTC Applications SDP. Issue needs to be worked by SE&I).

2. CSCI ASSESSMENTS

2.1 APPLICATIONS SERVICES CSCI (BASIC SERVICES)

Applications Services (Basic) Work Required

Tasks to be performed for this thread include:

1. Supply data related to Atlas delivered FD and Gateway services design.
2. Develop additional Application Services per Application Software API Requirements Matrix.
3. Work with EIM Services to explore alternative implementations of FD Services and Subsystem Services potentially necessary for EIM Support. Study alternative FD Services/ Subsystem Services classes.
4. Add support for String Type FDs.
5. (info only – this will be worked as part of the DDE Phase 2 Thread) Add DDE support for ControlShell™. Integrate Control Shell 6.0 application into Desktop Debug Environment. Provide DDE things necessary for Control Shell to get/put data through the model. Control Shell application expertise to be provided by EIM Services group.
6. (info only – this will be worked as part of the Commanding and CPRO Phase 3 Thread) Provide Application Services interfaces for command functionality delivered during Atlas.

ASV (Basic) Assessment

CSC Name	CSC Labor	% of CSC
ASV - study	3 LM	
ASV - user documentation	2 LM	
ASV - String type FD	3 LM	

Basis of estimate

Command & CPRO Phase 3 Thread handles the impacts from the new commands. Desktop Debug Environment (DDE) Phase 2 Thread handles impacts for ControlShell into DDE.

Documentation

Document Type	New/Update	Number of Pages
Requirements and Design Documentation	X	8
Users Guide	X	60
API Interface Document	X	6
Interface Design Document	X	10
Test Procedure	X	20

Other

We will provide an operating guide supplement to the DDE handbook that details Control-Shell operation in the DDE.

Assumptions

Specific performance data required will be gathered/instrumented/documented by another group.

Open Issues

Organize Application Services APIs such that it is possible to implement FD-specific and gateway-specific methods in separate ControlShell ™ components.

2.2 APPLICATIONS SERVICES CSCI (TAILORED SERVICES)

End Item Manager Services CSC Work Required

This CSC needs to provide Control Shell atomic components for approximately 300 application services, subsystem services and other services.

Prerequisite Control Logic Services CSC Work Required

This CSC needs to provide support for data types to be supported in Atlas.

Test Application Script Services CSC Work Required

This CSC needs to provide methods for writing, editing, running and debugging Test Application Scripts

CSCI Assessment

CSC Name	CSC Labor (LM)	% of CSC
End Item Manager Services	48	50% (development completion)
Prerequisite Control Logic Services	12	20% (development completion)
Test Application Script Services	24	100% (development completion)

Basis of estimate

Approximately 300 Control Shell atomic components at approximately 3 per person per week for EIM Services, and approximately 3.0 KSLOC at 2 lines/hour for PCL services and approximately 6.0 KSLOC at 2 lines/hour for TAS services, plus test support.

Documentation

Document Type	New/Update	Number of Pages
API Interface Document	Update	~10
Test Procedure	Update	~20

Assumptions

ControlShell™ 6.0 production release by April 17.

Open Issues

Onboard Services communications definition.

2.3 COMMAND SUPPORT CSCI

Command Management CSC Work Required

- Authenticates command requests utilizing an authentication interface.
- Interfaces with a PCL interface to determine FD commandability.

FD Commanding CSC Work Required

- Provide access to command FDs. Upgraded API to be delivered for Atlas.

Authentication Interface

- Provide an authentication interface for use in the CORBA filter. API to be delivered during Atlas.

CSCI Assessment

The “Commanding and Command Processor Thread” has this work identified in its thread. No additional work is identified.

CSC Name	CSC Labor (LM)	% of CSC
Command Processor	0	0
Command Management	0	0

Basis of estimate

Support for CCP Applications Interface Thread is included in Commanding and Command Processor Thread.

Documentation

Document Type	New/Update	Number of Pages
FD CMDing Interface Design Document	Update	*
Authentication API Interface Document	New	*
Test Procedure	New	*

* Included in “Commanding and Command Processor Thread”

Assumptions

- EIM commands to End Items will use the FD Services interface.
- EIM to EIM communications will be performed using CORBA interface
- No EIM commands will be entered through the manual command window, Command Processor. All EIM commanding will be performed through User Applications and the CORBA interface or through other EIMs and the CORBA interface.

Open Issues

- None.

2.4 SYSTEM CONTROL CSCI

System Control includes the Operation Configuration Manager (OPSCM) CSC that downloads baselines to all of the platforms in a test set and initializes processes in them.

Operation Configuration Manager (OPSCM) OPSCM CSC Work Required

OPSCM will initialize the EIM Initializer when the CCP on which the EIM will run is brought up. This is accomplished by the EIM Initializer being included in the startup script OPSCM runs once the SCID and TCID baselines have been downloaded to the appropriate CCP.

CSCI Assessment

Operation Configuration Manager	CSC Labor	% of CSC
Addition of EIM Initializer to startup script	0.25 MM	

Basis of estimate

Adding other processes to the OPSCM startup scripts.

Documentation

Document Type	New/Update	Number of Pages
Requirements and Design Documentation	Y	1
Users Guide	N	
API Interface Document	N	
Interface Design Document	N	
Test Procedure	Y	2

Other:

Assumptions

OPSCM must know where the EIM Initializer resides, the executable name and relevant Autopanel parameters in order to successfully start it.

Open Issues

None

2.5 SYSTEM VIEWERS CSCI

PCL Viewer CSC Work Required

As part of the System Viewers, a new FD Viewer will be defined, that will display the list of FDs that are associated to Prerequisite Control Logic. The display will include the following:

1. The list of FDs associated with PCL
2. For each FD, the PCL description (i.e., the PCL procedure name)
3. The capability to sort the display results based on FD Name.

The PCL FD Viewer will be invoked by the Status FD Viewer. Therefore, there is an additional (minor) impact to the Status FD Viewer.

CSCI Assessment

CSC Name	CSC Labor (LM)	% of CSC
PCL Viewer	5.0	

Basis of estimate

Adding a scrollable viewer with three control buttons.

Documentation

Document Type	New/Update	Number of Pages

Document Type	New/Update	Number of Pages
API Interface Document	Update	~10
Test Procedure	Update	~20

Assumptions

1. PCL FD Viewer will call Application Services (i.e., FD Services) to retrieve FDs associated with PCL.
2. As part of this API or another API, Application Services will return the PCL Description.
3. PCL FD Viewer will call Application Services to retrieve the current state (PCL-inhibit) from the CVT.
4. PCL FD Viewer will call Application Services to set/reset the inhibit flag of the respective FD. (Application Services has the responsibility to call Command for command authentication.)

2.6 COMMON APPLICATION SUPPORT

The Common Application Support (CAS) CSCI is the repository for RTC Application Software reusable components.

CAS Repository Work Required

The CAS CSCI will perform the following activities:

- Establish control procedures and practices for the introduction of ControlShell components into the repository and for their retrieval for use.
- Develop standardized interfaces between Sequencers and Test Application Displays for the common user actions (stop, start, resume, secure, terminate), prompt actions and report generation.

The development of the ControlShell components is not assessed by this CSCI. That development is performed during the RTC Application Software IPT activities.

CSCI Assessment

CSC Name	CSC Labor (LM)	% of CSC
Process Definition	2.0	
Sequencer Control Interfaces	6.0	
Total	**8.0**	

Basis of estimate

Documentation

Document Type	New/Update	Number of Pages
EIC Development Practice	New	~40
Component Interface Document	New	~2 each

Assumptions

1. The testing of the interface developed for the CAS repository will be performed as part of the HMF or Power Up/Down IPT validation (first use opportunities).

Open Issues

None.

3. HWCI ASSESSMENTS

None.

4. COTS PRODUCTS DEPENDENCIES

4.1 SW PRODUCTS DEPENDENCY LIST

Product Name	Quantity Needed	Need Date
ControlShell™ 6.0 Development License	25	3/27/98
ControlShell™ “Read Only” License	TBD	TBD
DOORS License	20	3/27/98

4.2 HW PRODUCTS DEPENDENCY LIST

None.